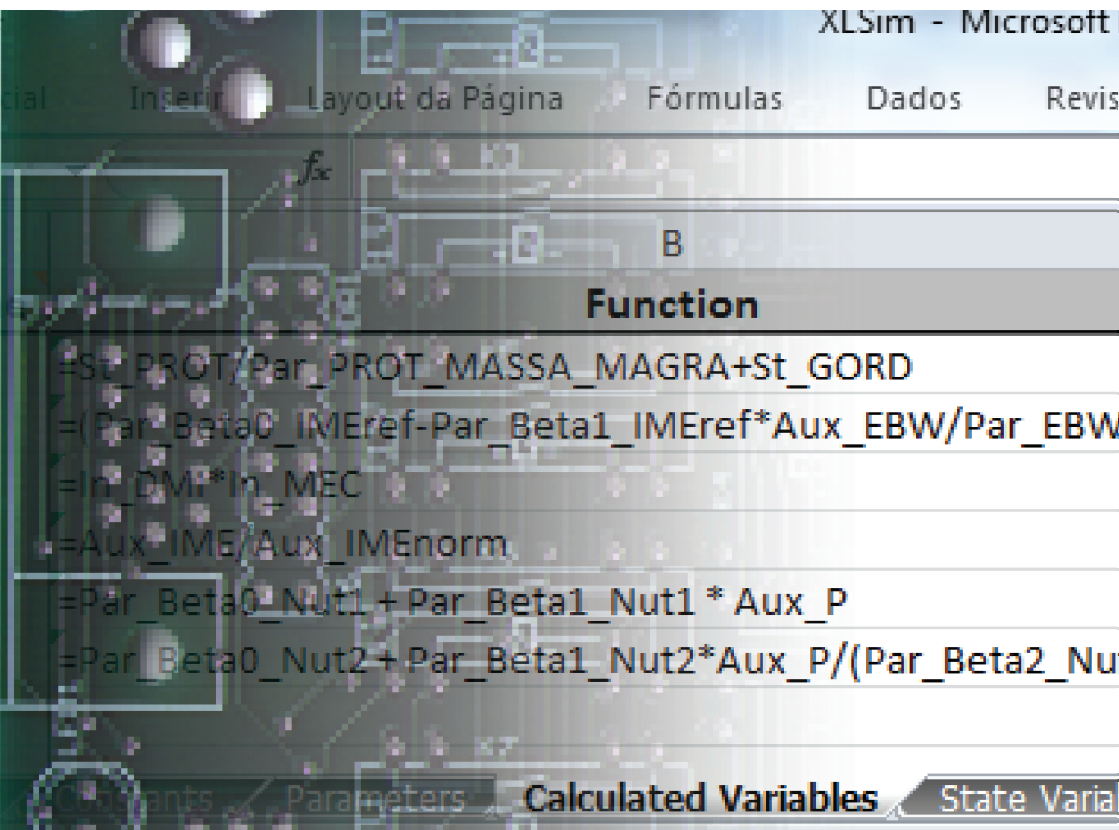


## Prototipação de Modelos e Geração Automática de Código-fonte em Planilha Eletrônica para o *Framework* de Modelagem MaCSim





*Empresa Brasileira de Pesquisa Agropecuária  
Embrapa Informática Agropecuária  
Ministério da Agricultura, Pecuária e Abastecimento*

# ***Boletim de Pesquisa e Desenvolvimento 35***

## **Prototipação de Modelos e Geração Automática de Código-fonte em Planilha Eletrônica para o *Framework* de Simulação MaCSim**

*Rodrigo Dias Ribeiro da Silva  
Luís Gustavo Barioni*

Embrapa Informática Agropecuária  
Campinas, SP  
2013

## **Embrapa Informática Agropecuária**

Av. André Tosello, 209 - Barão Geraldo  
Caixa Postal 6041 - 13083-886 - Campinas, SP  
Fone: (19) 3211-5700 - Fax: (19) 3211-5754  
[www.cnptia.embrapa.br](http://www.cnptia.embrapa.br)  
[cnptia.sac@embrapa.br](mailto:cnptia.sac@embrapa.br)

### **Comitê de Publicações**

Presidente: *Silvia Maria Fonseca Silveira Massruhá*

Secretária: *Carla Cristiane Osawa*

Membros: *Adhemar Zerlotini Neto, Stanley Robson de Medeiros Oliveira, Thiago Teixeira Santos, Maria Goretti Gurgel Praxedes, Adriana Farah Gonzalez, Neide Makiko Furukawa, Carla Cristiane Osawa*

Membros suplentes: *Felipe Rodrigues da Silva, José Ruy Porto de Carvalho, Eduardo Delgado Assad, Fábio César da Silva*

Supervisor editorial: *Stanley Robson de Medeiros Oliveira, Neide Makiko Furukawa*

Revisor de texto: *Adriana Farah Gonzalez*

Normalização bibliográfica: *Maria Goretti Gurgel Praxedes*

Editoração eletrônica/Capa: *Neide Makiko Furukawa*

Imagens da capa: <http://tiqx.blogspot.com.br/2012/05/vga-switch-um-projeto-da-canonical.html>, planilha elaborada pelos autores

### **1ª edição**

on-line 2013

#### **Todos os direitos reservados.**

A reprodução não autorizada desta publicação, no todo ou em parte, constitui violação dos direitos autorais (Lei nº 9.610).

#### **Dados Internacionais de Catalogação na Publicação (CIP)**

Embrapa Informática Agropecuária

---

Silva, Rodrigo Dias Ribeiro.

Prototipação de modelos e geração automática de código-fonte em planilha eletrônica para o *framework* de modelagem MaCSim / Rodrigo Dias Ribeiro da Silva, Luis Gustavo Barioni. - Campinas : Embrapa Informática Agropecuária, 2013.

13 p. : il. - (Boletim de pesquisa e desenvolvimento / Embrapa Informática Agropecuária, ISSN 1677-9266 ; 35).

1. Modelo matemático. 2. Framework de simulação. 3. Modelo e simulação. I. Barioni, Luis Gustavo. II. Embrapa Informática Agropecuária. III. Título. IV. Série.

# Sumário

<b>Resumo .....</b>	<b>5</b>
<b>Abstract .....</b>	<b>7</b>
<b>Introdução .....</b>	<b>8</b>
<b>Material e Métodos .....</b>	<b>8</b>
<b>Resultados e Discussão .....</b>	<b>10</b>
<b>Conclusões .....</b>	<b>12</b>
<b>Referências .....</b>	<b>13</b>



# Prototipação de Modelos e Geração Automática de Código-fonte em Planilha Eletrônica para o *Framework* de Simulação MaCSim

---

Rodrigo Dias Ribeiro da Silva<sup>1</sup>

Luís Gustavo Barioni<sup>2</sup>

## Resumo

O Laboratório de Matemática Computacional (LabMaC) da Embrapa Informática Agropecuária vem desenvolvendo um *framework* para simulação orientada a objetos, implementado na linguagem C++, o MaCSim. Uma das desvantagens de *frameworks* orientados a objetos é a sua baixa expressividade matemática e visual, dificultando a interação com especialistas de domínio e exigindo destes especificação matemática ou textual do modelo antes da implementação por programadores. Tal processo é moroso e sujeito a erro. Planilhas eletrônicas são ferramentas com as quais especialistas de domínio, em geral, possuem grande familiaridade. Nesse contexto, decidiu-se por desenvolver um software em planilha eletrônica que permita a prototipação de modelos de dinâmica de sistemas. Tal software permite a especificação e a execução do modelo em plani-

---

<sup>1</sup> Graduando em Engenharia agrícola, estagiário da Embrapa Informática Agropecuária, Campinas, SP, rodrigo.drds@gmail.com

<sup>2</sup> Engenheiro agrônomo, Doutor em Ciência animal e pastagens, pesquisador da Embrapa Informática Agropecuária, Campinas, SP, luis.barioni@embrapa.br

lha, além da gravação e da recuperação da especificação do modelo em arquivos texto e geração de código fonte C++ compatível com o *framework* MaCSim. O prototipador de modelos aqui descrito é uma ferramenta de rápido aprendizado por ter como base uma planilha eletrônica, comumente utilizada por pesquisadores em geral. Apesar da geração de código ainda não ter sido totalmente automatizada, a ferramenta tem se mostrado útil para agilizar o desenvolvimento de modelos matemáticos da dinâmica de sistemas, reduzir a frequência de erros e facilitar o processo de comunicação e especificação do modelo, em conjunto com especialistas de domínio. Esse artigo tem por objetivo descrever as características básicas de uma ferramenta de prototipação e geração automática de código-fonte C++ para modelos de dinâmica de sistemas.

**Termos para indexação:** *Framework* de simulação, modelos matemáticos, planilha eletrônica.



# A Spreadsheet for Model Prototyping and Automatic Source Code Generation for the MacSim Simulation Framework

---

## Abstract

*The Computational Mathematics Laboratory of the Embrapa Agricultural Informatics research center is developing an object-oriented simulation framework implemented in C++, named MaCSim. However, a disadvantage of an object-oriented framework, such as MaCSim, is that coding a model in C++ has low visual and mathematical expressiveness, impairing the interaction with domain experts. Consequently, it would require a textual specification of the model from the domain expert before its implementation by programmers. Also, the compiled program should be sent back to the domain expert for testing. This process is inefficient and error prone. Electronic spreadsheets are familiar tools for domain experts in general. Therefore, we have decided to develop a spreadsheet-based software (XLSim) which supports prototyping system dynamic models. XLSim allows to specify the model in a standard format and run it in the spreadsheet, besides allowing saving the specification in text files and generating C++ source-code which is compatible with the MaCSim framework. XLSim is a tool which is easy to learn thanks to its basically standard spreadsheet interface. Code generation not totally automatic because it does not support all the “built in” spreadsheet functions. In those cases, the code generator comment the line of code with the unknown function and requires action from a computer programmer. However, XLSim has been evaluated as a very useful tool to speed-up the development and to ,minimize the frequency of specification and implementation errors, besides facilitating the communication with domain experts. This article has the objective of describing the basic characteristics of a spreadsheet-based system dynamics model prototyping and code generator tool.*

**Indexterms:** *Simulation Framework, mathematical models, electronic spreadsheet.*

## Introdução

O Laboratório de Matemática Computacional (LabMaC) da Embrapa Informática Agropecuária vem desenvolvendo um *framework* para simulação orientada a objetos denominado MacSim, implementado na linguagem de programação C++. O *framework* foi projetado para dar suporte, inicialmente, à implementação de modelos baseados em equações diferenciais de primeira ordem utilizando a abordagem de dinâmica de sistemas.

Uma das maiores desvantagens da implementação de simuladores em linguagens de programação utilizadas em *frameworks* orientados a objetos é a sua baixa expressividade matemática e visual, o que dificulta a interação com especialistas de domínio (IBA et al., 2004). Ainda a especificação de modelos (i.e. a definição de variáveis, parâmetros, equações, etc...) de forma textual possui desvantagens. Como tal, a especificação textual não pode ser convenientemente verificada e testada antes que um programador codifique o modelo adequadamente e possa submeter o software implementado para que os resultados do modelo possam ser verificados por um especialista de domínio (RAHMAN; MIZUKAWA, 2013). Nesse processo, há grande propensão a erros de especificação por parte do especialista de domínio e morosidade no processo de desenvolvimento e testes. Além disso, programadores menos experientes no uso do *framework* estão, também, sujeitos a erros na implementação do modelo nos padrões requeridos pelo *framework*.

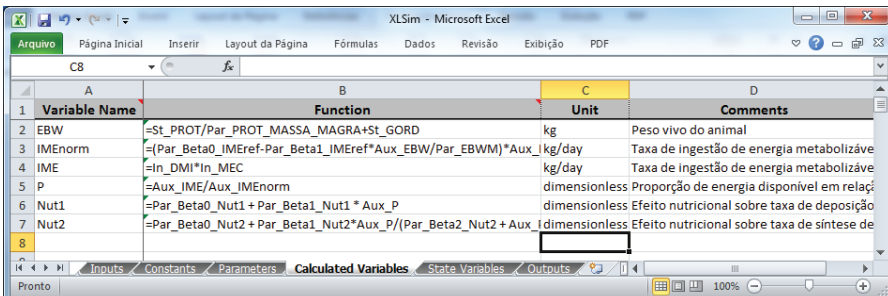
Nesse contexto, esse trabalho tem por objetivo descrever as características básicas de uma ferramenta de prototipação de modelos em planilha eletrônica capaz de gerar código-fonte em C++ para um *framework* de simulação orientado a objetos.

## Material e Métodos

Planilhas eletrônicas são ferramentas com as quais especialistas de domínio, em geral, possuem grande familiaridade. Assim, decidiu-se por desen-

volver um software em planilha eletrônica que permite a prototipação de modelos de dinâmica de sistemas. Tal software permite a especificação, de forma padronizada, e a execução do modelo, permitindo o salvamento da especificação em arquivos texto e a geração de código fonte C++ compatível com o *framework* MaCSim.

Na planilha, um modelo matemático é especificado definindo-se variáveis e equações em um *template* (Figura 1) utilizando-se abas pré-estabelecidas na planilha. Cada aba agrupa um “tipo” de variável, seguindo o padrão clássico de modelos implementados utilizando-se o paradigma de sistemas dinâmicos (parâmetros, constantes, variáveis de estado, variáveis auxiliares, inputs e outputs). A planilha nomeia automaticamente as células contendo o valor das variáveis com o nome correspondente especificado na célula imediatamente à sua esquerda. Dessa forma, as equações do modelo são especificadas com o nome da variáveis utilizadas na fórmula e não com os endereço da células que contém os valores necessários. A especificação dos valores e das funções tendo como base os nomes das variáveis, além de promover uma maior facilidade de leitura da planilha, é essencial para a geração da documentação textual e do código-fonte em C++ utilizando-se dos mesmos nomes de variáveis definidos na planilha.



Variable Name	Function	Unit	Comments
EBW	=St_PROT/Par_PROT_MASSA_MAGRA+St_GORD	kg	Peso vivo do animal
IMEnorm	=(Par_Beta0_IMEref-Par_Beta1_IMEref*Aux_EBW/Par_EBWM)*Aux	kg/day	Taxa de ingestão de energia metabolizável
IME	=In_DMI*In_MEC	kg/day	Taxa de ingestão de energia metabolizável
P	=Aux_IME/Aux_IMEnorm	dimensionless	Proporção de energia disponível em relação
Nut1	=Par_Beta0_Nut1 + Par_Beta1_Nut1 * Aux_P	dimensionless	Efeito nutricional sobre taxa de deposição
Nut2	=Par_Beta0_Nut2 + Par_Beta1_Nut2*Aux_P/(Par_Beta2_Nut2 + Aux	dimensionless	Efeito nutricional sobre taxa de síntese de

**Figura 1.** Exemplo de especificação do modelo em planilha.

Equações diferenciais e equações auxiliares são definidas por meio de formulas com a sintaxe padrão do MS Excel v15 (Office 365). Para execução da simulação foi implementado o algoritmo de Runge-Kutta (PRESS et al., 1992) com malha de tempo homogênea (i.e. passo de tempo fixo). Variáveis de controle da simulação (e.g. tempo de simulação, passo de

tempo da malha, passo de tempo para gravação de saídas, etc.) são definidas em uma aba específica da planilha chamada *Simulation*.

## Resultados e Discussão

O modelo especificado na planilha é interpretado por meio de rotinas escritas em Visual Basic para Aplicativos (VBA), acionadas por meio de botões incluídos na interface da planilha (Figura 1). O software de planilha implementa também rotinas para verificação de erros de especificação e conformidade. Essas rotinas incluem, por exemplo, a verificação de: (1) presença de valores de variáveis definidos por endereço nas fórmulas, (2) presença de variáveis de saída nas fórmulas do modelo, (3) adequação dos valores dos parâmetros e das variáveis à faixa de amplitude válida especificada. Toda a funcionalidade do MS Excel permanece disponível ao usuário, exceto a criação de novas abas e gráficos. Dados de saída podem ser copiados para outra planilha para análise.

Arquivos com texto, descrevendo o modelo podem ser gerados pela planilha. O arquivo texto tem duas finalidades: (a) armazenar a especificação do modelo matemático e permitir que ele seja novamente importado para a planilha, evitando-se dessa forma a necessidade de gravar um arquivo de planilha para cada modelo, e; (b) permitir a visualização da especificação do modelo em uma forma mais compacta do que a proporcionada pela planilha. Além disso, o XLSim gera arquivos de cabeçalho (\*.h) e implementação (\*.cpp) em código C++, compatível com o *Framework* MaCSim. As funções de planilha utilizadas na especificação são convertidas à função equivalente em C++, com suporte para as versões em português e inglês do Microsoft Excel. São suportadas funções matemáticas comuns como funções trigonométricas, logarítmicas, exponenciais, potências e raízes.

A Figura 2 apresenta um exemplo de especificação textual gerada pela planilha. Na especificação textual, o nome do modelo, os parâmetros com

```

@Model:0ltjen

@Inputs

In_DMI = 8
In_MEC = 3
In_NEm = 1,8
In_Neg = 1,2

@Parameters

Par_PROT_MASSA_MAGRA = 0,2201
Par_EPROT = 5,539
Par_EGORD = 9,385
Par_Beta0_IMEref = 0,438
Par_Beta1_IMEref = 0,2615
Par_EBMWref = 750
Par_EBWM = 1050
Par_Beta0_Nut1 = -0,7
Par_Beta1_Nut1 = 1,7
Par_Beta0_Nut2 = 0,83
Par_Beta1_Nut2 = 0,2
Par_Beta2_Nut2 = 0,15
Par_k1 = 0,00429
Par_k2 = 0,0461
Par_k3 = 0,143
Par_DNAmx = 385
Par_alfa = 0,0858

@Calculated Variables

Aux_EBW=St_PROT/Par_PROT_MASSA_MAGRA+St_GORD
Aux_IMEnorm=(Par_Beta0_IMEref-Par_Beta1_IMEref*Aux_EBW/Par_EBWM)*Aux_EBW^0.73
Aux_IME=In_DMI*In_MEC
Aux_P=Aux_IME/Aux_IMEnorm
Aux_Nut1=Par_Beta0_Nut1 + Par_Beta1_Nut1 * Aux_P
Aux_Nut2=Par_Beta0_Nut2 + Par_Beta1_Nut2*Aux_P/(Par_Beta2_Nut2 + Aux_P)

```

**Figura 2.** Exemplo de especificação textual produzida pela planilha.

seus respectivos valores, equações diferenciais, valores de estado inicial e equações auxiliares são definidas a partir das variáveis especificadas.

A planilha gera código de uma classe especializada a partir da classe *Model* do *framework* MaCSim. As variáveis são declaradas, conforme seu tipo (i.e. variáveis de entrada, constantes, parâmetros, variáveis de estado, etc.) e as equações implementadas na planilha são convertidas em métodos da classe modelo do *framework* MaCSim, as quais são utilizadas como ponteiros para avaliação do valor das variáveis. A Figura 3 apresenta um trecho de código-fonte gerado pela planilha.

```

double Cls_Oltjen_Model::d_dt_DNA(){
    return Par_k1->Value()*Par_DNAmax->Value()
        - St_DNA->SearchOutputPort("Stock")->Value()*Aux_Nut1->Value();
}
double Cls_Oltjen_Model::d_dt_PROT(){
    return Aux_Nut2->Value()*Par_k2->Value()*St_DNA->SearchOutputPort("Stock")->Value()^(0.73)
        - Par_k3->Value()*St_PROT->SearchOutputPort("Stock")->Value()^(0.73);
}
double Cls_Oltjen_Model::d_dt_GORD(){
    return ((In_DMI->Value()-Par_alfa->Value()*Aux_EBW->Value()^0.73/In_NEM->Value())*In_Neg->Value()
        - St_PROT->SearchOutputPort("Stock")->Value()*Par_EPROT->Value()/Par_EGORD->Value());
}

```

**Figura 3.** Fragmento de código-fonte em C++ produzido pela planilha.

## Conclusões

O prototipador de modelos aqui descrito é uma ferramenta simples, de rápido aprendizado por ter como base uma planilha eletrônica, comumente utilizada por pesquisadores em geral. Até o presente momento a experiência com a planilha é restrita, mas pesquisadores que já puderam utilizá-la, reportam boas experiências em termos de usabilidade e funcionalidade.

As primeiras experiências com a geração de código automático indicam que o software de planilha é muito promissor. Há ainda a limitação, entretanto, de que a conversão de diversas funções “Built in” do MS Excel para C++ ainda não é totalmente automática. Quando o interpretador não é capaz de converter a função de planilha menos usual ou uma operação matemática especificada, a função é comentada no código gerado e há necessidade de intervenção de um programador. Em geral, as intervenções de programadores tendem a ser raras.

Apesar da geração de código ainda não ter sido totalmente automatizada, o prototipador de modelos em planilha, se mostra útil para agilizar o desenvolvimento, reduzir a frequência de erros e facilitar o processo de comunicação e especificação do modelo em conjunto com especialistas de domínio no processo de implementação de modelos com o *Framework* MaCSim.

## Referências

IBA, T.; MATSUZAWA, Y.; AOYAMA, N. From conceptual models to simulation models: model driven development of agent-based simulations. In: WORKSHOP ON ECONOMICS AND HETEROGENEOUS INTERACTING AGENTS, 9., Kyoto, 2004. **Proceedings...** Kioto: Kioto University, 2004. (Lectures Notes in Economics and Mathematical Systems, 567). Disponível em: <<http://platbox.sfc.keio.ac.jp/en/papers/wehia2004/box-wehia2004.pdf>>. Acesso em: 31 out. 2013.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical recipes in C: the art of scientific computing**. 2nd ed. Cambridge: Cambridge University, 1992. 994 p.

RAHMAN, M. A. A.; MIZUKAWA, M. Modeling and design of mechatronics system with SysML, simscape and simulink. In: IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, Wollongong. **Proceedings...** Piscataway: IEEE; Wollongong: University of Wollongong, 2013. p. 1767-1773.



---

*Informática Agropecuária*

Ministério da  
**Agricultura, Pecuária  
e Abastecimento**



CGPE 11171